Matt Winkler

Physics 398EMI

Final Project Report

“A MIDI Controlled IC Synthesizer”

December 20, 2002

The History of Synthesizers

The first electronic musical instrument was the Telharmonium, which was patented in 1896 by Thadeus Cahill. The Telharmonium was designed to broadcast music into homes, restaurants, and stores via telephone lines. This pioneer instrument weighed close to 200 tons because it used dynamos to generate sound.

The next step in the evolution of the synthesizer is the Theremin, which was developed by Leon Theremin in Russia in the 1920. The Theremin utilizes antennae that detect the proximity of the player's hands. The pitch and volume of the output sound is modulated according to the signals from the antennae.

Not far behind the Theremin was Laurens Hammond's "Hammond Organ," which became commercially available in 1935. This organ and others like it influenced the radio industry of the 1940's. Much of the background music for radio dramas was being created electronically via these instruments. This was one of the first ways the general public was exposed to any sort electronic music.

In 1957, RCA built the RCA Mark II Electronic Music Synthesizer, which was an analog synthesizer (vacuum tube based) that was controlled with punched paper tape. RCA set out to create a synthesizer that could create virtually any sound imaginable. The Mark II, however, was far too complex for the general public and was never commercially sold.

In 1964, Robert Moog developed his first synthesizers and totally reshaped the way synthesizers were designed, used, and thought of. Moog's synthesizers were modular and voltage-controlled. The various modules (oscillators, filters, mixers, amplifiers, envelope generators, etc.) were connected with patch cables and were controlled with keyboards. This precursor to "plug and play" technology allowed musicians to shape the sound of their synthesizers to fit their needs. Never before had synthesizers been thought of as musical instruments with "voices" to rival those of traditional instruments.

From 1964 on, the synthesizer industry took off. Digital synthesizers began to come about in the 1970's and have since become the industry standard. Today, there is a renewed interest in the sound of the old analog synths as well as a push for new synthesizer technology. (DSP synths and the like)

The History of MIDI

Some early commercial analog synthesizers utilized an analog control interface to send pitch information and note on/off events and to provide a timing reference in-between synthesizers. These interfaces, however, were not standardized and varied from manufacturer to manufacturer. (For example, different manufacturers used different voltage to pitch schemes and used different clock speeds for the timing references.) Also, these early interfaces only allowed for one note at a time to be controlled. Musicians using synthesizers from different manufacturers had to learn the differences between the various interfaces and had to adapt their work to fit within the constraints of each system. A definite need had arisen for a standardized control system that allowed for interconnectivity.

By the early 1980's, commercial synthesizers had become extremely complex. Microprocessors and various other digital elements were finding their way into commercial synthesizers because digital components were smaller and more reliable than analog components and the production cost of integrated circuits was at an all time low. This "digital revolution" in synthesizer design led to a "digital revolution" of synthesizer control.

This digital revolution was MIDI 1.0. MIDI 1.0 was the brainchild of audio engineers from Sequential Circuits, Roland Corporation, and Oberheim Electronics. This groundbreaking system of standards was published in August of 1983. The MIDI (Musical Instrument Digital Interface) standard specifies connections at both the hardware and communications protocol levels. (All physical MIDI connections use the same cable and all MIDI devices use the same language to communicate.)   These two simple points illustrate the true power of the MIDI protocol.

MIDI has been improved somewhat since 1983. The newer improvements include control over musical features like pitch bending, slurring, tremolo, and reverb. (Any new changes to the protocol must be passed by the MIDI Manufacturers' Association (MMA) and the Japan MIDI Standards Committee.) The actual meat and potatoes of the protocol, however, has not changed a bit. MIDI synthesizers from 1985 still respond exactly as they are supposed to to MIDI messages sent from today's latest-and-greatest sequencers and controllers.

The Original Project Plan

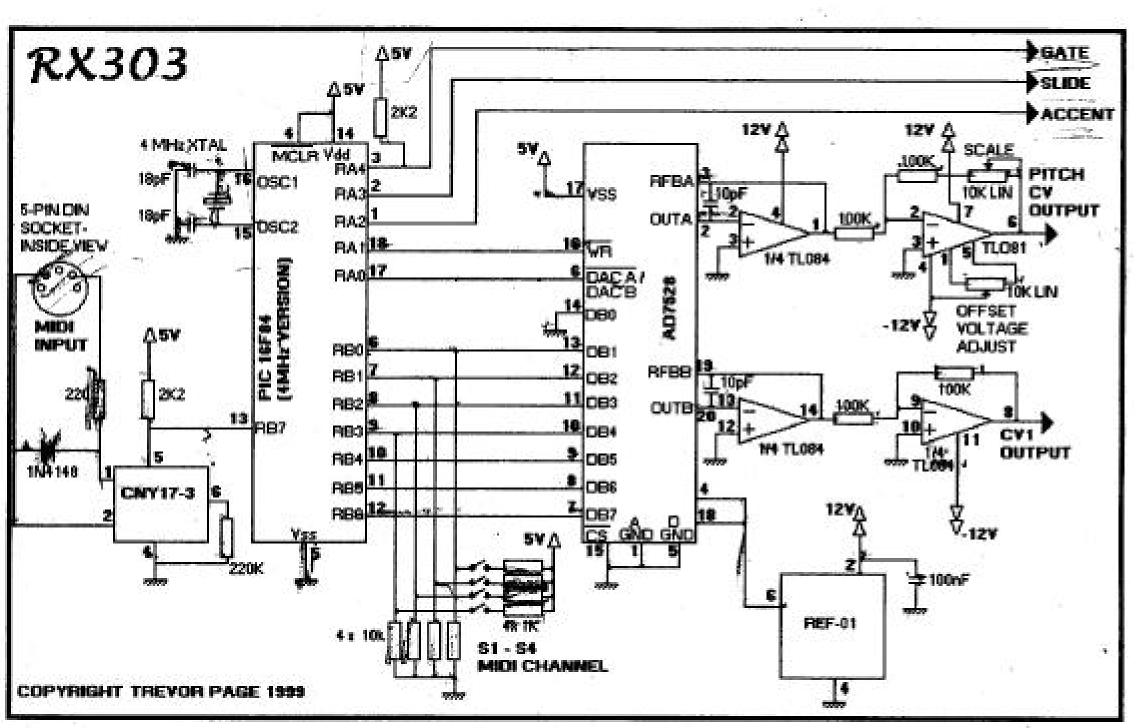
At the beginning of the semester, I planned on building a simple, conventional, digital synthesizer using a MIDI to CV (control voltage) schematic that I found on the web and a function generator IC. The IC I planned to use was the Maxim MAX-038. The MAX-038 outputs a sine, square, or triangle wave at a frequency that is determined by the input current. The synthesizer should create tones when MIDI information is sent to it by an outside MIDI controller like a keyboard or a sequencer.

What I Actually Built

What I actually ended up building/designing was close to the original project plan. I did, in fact, use the MIDI to CV schematic that I found on the web and the MAX-038. There were a few instances, however, in which the final product does not exactly match the plan. I will explain those differences later. First, I will explain how the circuit works and the steps that I went through to design and build my synthesizer.

The MIDI to CV converter schematic that I used was designed by Trevor Page. Trevor also wrote the code that was downloaded onto the PIC16F84 microcontroller. Without Trevor's design, code, and helpful e-mails, this project would have been much more impossible than it already was. This circuit, the

RX303, was designed to provide MIDI capability to the world renowned Roland TB303 synthesizer. The TB303 was a pre-MIDI 1.0 machine but was equipped with an analog control interface like the ones I described above. The RX303 MIDI to CV converter was meant to translate MIDI messages to the “language” of the TB303’s control interface. Below is the RX303 MIDI to CV circuit as originally created by Trevor Page.



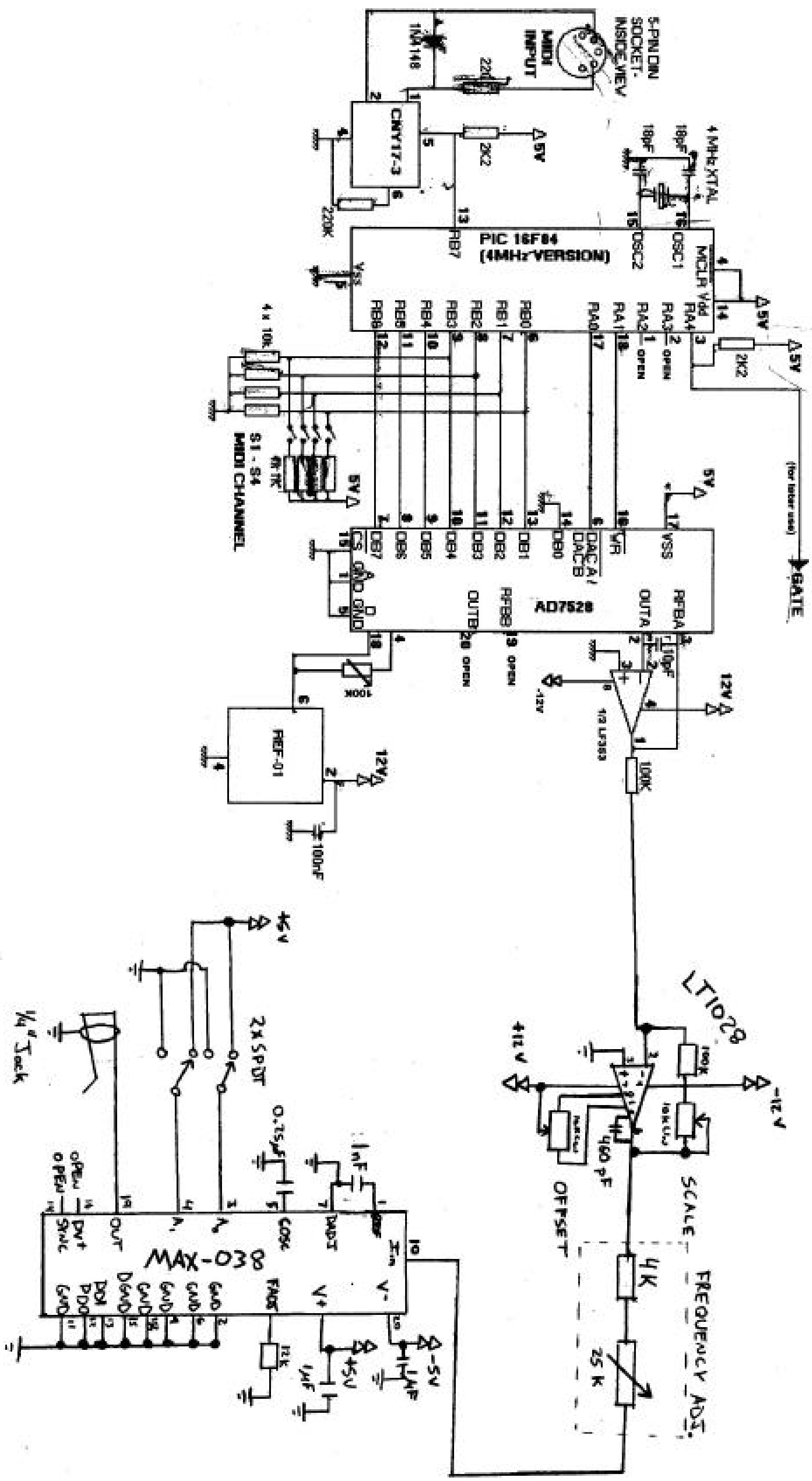
The circuit first receives MIDI messages from a controller and sends them to the CNY17-3 optocoupler. The optocoupler translates the incoming MIDI information to TTL 1’s and 0’s that the PIC16F84 can understand.

The PIC16F84 microcontroller stores the necessary code and follows out the instructions in the code in accordance to the signals sent to it by the optocoupler. Getting the code onto the PIC16F84 was my first major hurdle in getting my synthesizer to work. After talking to many people over the course of a few weeks and a couple failed attempts, I ended up talking to Michael Kasten, Michael Haney, and Todd Moore; three research engineers for the physics department. With their help I was finally able to download the code onto my PIC.

The PIC produces TTL-level signals on the Gate, Slide, and Accent outputs when triggered. Gate is triggered when a note is pressed. Accent is triggered when a note's velocity (volume) exceeds 100 (MIDI velocities are measured on a scale from 0 to 127). Slide is triggered when a second note is pressed while another is being held down (slide is used to mimic glissando). The switches connected to the logic outputs of the PIC are used to select the MIDI channel that the circuit will respond to. There are 16 MIDI channels, each one corresponding to a series of ons and offs selected on the four switches. Those same logic outputs carry the digital signals encoded for specific voltage levels that correspond to the pitch of the note that is pressed.

These digital signals are sent to the AD7528 dual 8-bit DAC. The DAC does as a DAC should and creates analog voltages at its outputs corresponding to the digital information coming in. The AD7528 uses a REF-01 precision voltage reference to create these analog voltages. One of the DACs on the AD7528 outputs a voltage corresponding to the pressed note's pitch and the other outputs a voltage corresponding to any other parameter that one would like to write into the code stored on the PIC. The op-amps at the outputs of the AD7528 act as buffers and allow for fine-tuning of the output signals.

While designing my synthesizer, I made some minor changes to the original RX303 circuit. I also had to tailor the various features of the MAX-038 function generator to suit my needs. The schematic for my whole project is below. It shows the modified RX303 circuit plus the function generator.


5-PIN DIN SOCKET-INSIDE VIEW
MIDI INPUT
1N4148
CNY17-3
4 MHz XTAL
PIC 16F84
(4MHz VERSION)
(for later use)
GATE
S1 - S4
MIDI CHANNEL
AD7528
REF-01
1/2 LF353
LT1028
SCALE
OFFSET
FREQUENCY ADJ.
MAX-038
2xSPDT
1/4" Jack

The first change to the RX303 circuit that I made was the addition of a 100K? pot between the voltage reference and the reference-in pin on the AD7528 for the pitch DAC. This allowed me to fine tune the output of the DAC without compromising the output signal. Because the output of the DAC relies directly on the level of the voltage reference, lowering the voltage supplied by the voltage reference lowered the level of the DAC's output. This was necessary to keep the MIDI to CV converter's output current at a lower level to protect the MAX-038 function generator.

I also deleted the slide and accent outputs and the op-amps for the CV1 output. I decided that I would not need slide and accent or a control voltage for any other parameter. In addition, instead of using one-fourth of a TL084 quad op-amp for the DAC output buffer I used half of a LF353 dual op-amp and instead of using a TL081 precision op-amp I used a LT1028 precision op-amp on account of availability.

The Maxim MAX-038 outputs a frequency that depends on the input current. When FADJ is disabled, the output frequency is given by:

$$F_{out}(MHz) = \frac{I_{in}(mA)}{2C_F(pF)} \quad \text{or} \quad F_{out}(Hz) = \frac{V_{in}(V)}{2C_F(F) \times R_{in}(\Omega)}$$

where F_{out} is the output frequency, I_{in} is the input current, C_F is the capacitance of the capacitor between COSC and ground, V_{in} is the voltage on the input, and R_{in} is the resistance between the output of the RX303 and the input of the MAX-038. Because I_{in} must not exceed 750 µA, the components used with the MAX-038 were carefully selected to stay within this constraint while allowing a large frequency range.

The SPDT switches connected to A0 and A1 are used to select the waveform that is output by the MAX-038 (sine triangle, or square). The phase detection, external frequency synchronization, output frequency modulation

(FADJ), of the MAX-038 were disabled. The pot between the RX303 circuit and the MAX-038 is used for frequency adjustments. This will be explained later.

Most of the circuit was laid out on a breadboard before soldering it together. This aided in debugging, though the worst problems I had at that stage were accidental shorts and worn-out breadboards. My project is currently enclosed in the chassis from a non-functioning Spectra Physics exciter that Professor Errede and I found on the loading dock of Loomis. For power, I am using a computer power supply from an old computer in my basement. This was a very good find because it provides +5V, -5V, +12V, and –12V; every supply that I needed for my circuit.

That Which Strayed From the Original Plan

Though from the preceding explanation it may seem like my project has gone exactly as planned, it did not. The synthesizer that I built is by no means conventional. First of all, my synthesizer outputs a tone (at line-level) even if no keys are being pressed on the keyboard. Also, the musical scale is exponential and the scale on my synthesizer is linear. These two oddities make my synthesizer seem more like a MIDI-responsive function generator than a musical instrument, but it does make some interesting sounds that I believe are musically useful.

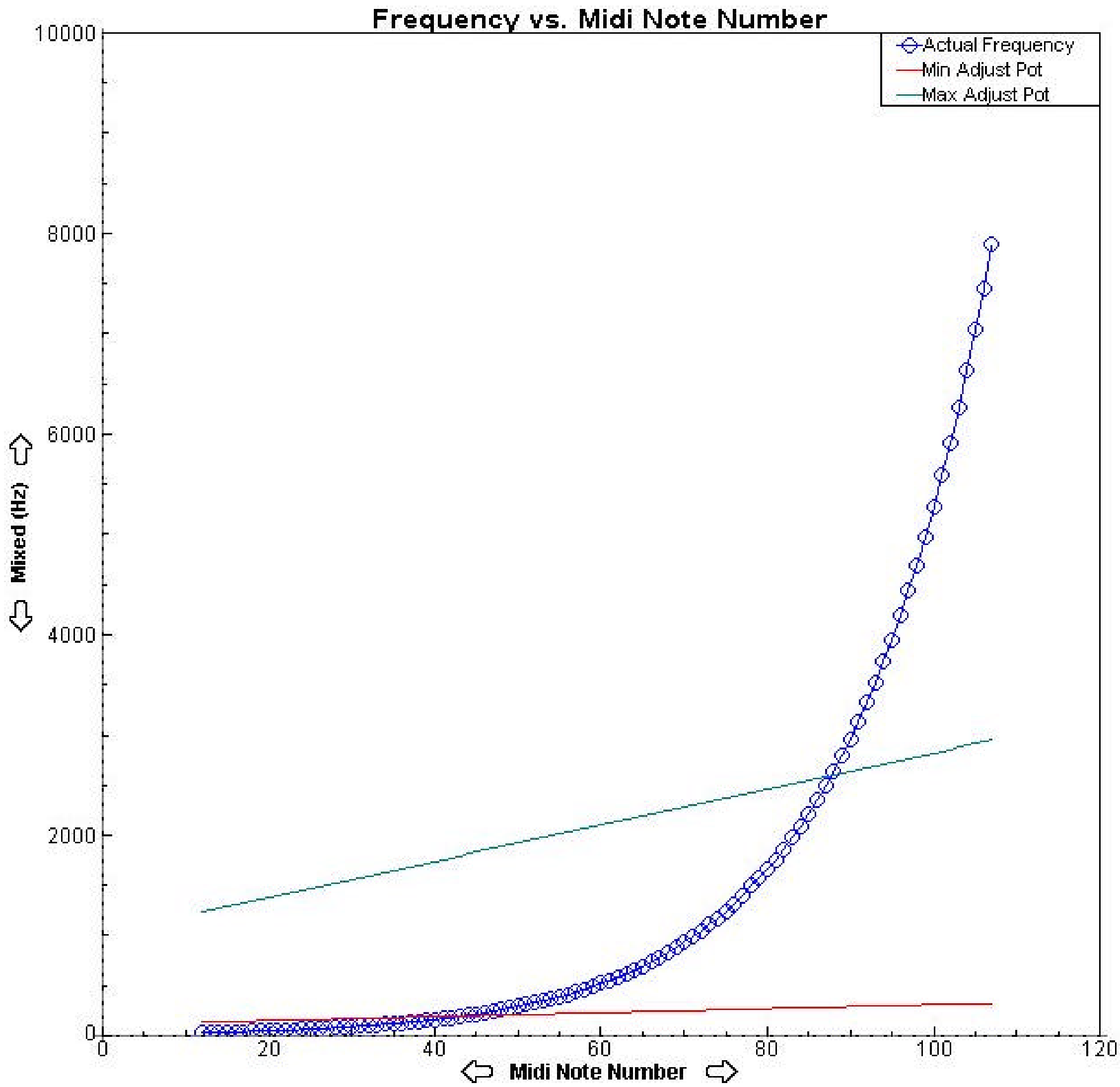
In the beginning, I did not understand what the gate output on the PIC did and I believed that the RX303 would only output a voltage if a note was being pressed. I was very wrong. The gate output is the only output that responds to note-on and note-off messages (5V for on 0V for off). I was unsuccessful in my several attempts to make a voltage-controlled switch to control the output voltage using various types of FETs. I also considered the use of a LDR (light dependant resistor) switch but the gate output does not carry enough current to operate one. At the moment, the only feasible option I can see that will fix my problem is to use a variable gain amplifier IC like the AD600. Because of time constraints, my synthesizer will not have one of these until after the end of the semester and will, for now, always be in a note-on state.

The exponentiality of the musical scale is also a great hurdle for me. I also naively believed that because the musical scale was exponential, the output of the RX303 circuit would be exponential to match it. Only God knows why a company as innovative as Roland would base a pitch control interface for a musical instrument on a linear scale. To compensate for this, I looked into several antilog amplifier designs using everyday op-amps. These designs work very well in an ideal environment, but the extreme temperature dependence of the components involved and the limit on current that is imposed by the MAX-038 renders these designs useless. I have been told that there are IC's that do a good job of exponentiating, but again, because of time constraints, I plan on looking into this after the semester is over. On the bright side, the linear response of my synthesizer allows me to experiment with tones in-between the semitones that conventional music theory lays out.

The Frequency Adjust pot in my schematic helps my synthesizer become a little less linear. Changing the resistance alters the current input to the MAX-038. This allows for a wide range of output frequencies for a single note on the keyboard. Using the Frequency Adjust pot with notes within the normal musical scale, I can attain frequencies between 137 Hz ($C^{\#}_2$) and 2948 Hz ($F^{\#}_6$). This encapsulates the middle of the musical scale.

Below is a plot showing frequency vs. MIDI note number. For reference, C_4, middle C, is MIDI note number 60. The two straight lines represent my synthesizer's frequency vs. MIDI note number curves when the Frequency Adjust pot is at its minimum and maximum values. My synthesizer can reach any frequency in-between the two straight lines for a certain MIDI note number. The blue exponential curve is the actual frequency vs. MIDI note number curve. Also for reference, the actual frequency of a note in terms of its MIDI note number is given by:

$$Frequency(Hz) \approx 16.351 \exp(0.057765 \times MIDInoteNumber)$$



Acknowledgements

Many thanks go out to Trevor Page for designing the RX303 circuit, writing the code for the PIC, and for all the helpful e-mails. Thank you to Michael Kasten, Michael Haney, and Todd Moore for helping me get the code written to the PIC. Thank you to Professor Errede for helping with everything, and thank you to everyone in the class who helped me out with anything. This project would have been much more impossible without any one of you!